UNIVERSITÉ
DE LORRAINE

Approaching dialogue modeling in a dynamic framework

Master's Thesis

To be defended on June, 19th, 2017

in order to get

Master's degree of Université de Lorraine
(in computer science)

by

Maria Boritchev

Under the supervision of Maxime Amblard

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

From the tip of his wand burst the silver doe: she landed on the office floor, bounded once across the office and soared out of the window. Dumbledore watched her fly away, and as her silvery glow faded he turned back to Snape, and his eyes were full of tears.

‘After all this time?’

‘Always,’ said Snape.

— J. K Rowling, *Harry Potter and the Deathly Hallows*

Introduction

This study presents an approach to dialogue modeling in a dynamic framework. Studying dialogical interactions is a major issue in natural language processing, since dialogues compose the basis of human communication. Addressing this problem is a complex task linking approaches from fields such as semantics, pragmatics, and more generally logic and cognitive science. As Type Theoretical Dynamic Logic (TTDL) [de Groote, 2006] provides a dynamic framework allowing to handle discourse as a linear aggregation of propositions, we hereafter consider dialogues as linear aggregations of speech turns, each being a piece of discourse, which permits us to develop tools to handle specifics of dialogical interactions on top of already existing methods for general discourse.
Since 2012, Maxime Amblard et al. have been working on the SLAM – Schizophrenia and Language, Analysis and Modeling – project, aiming to systematize pathological conversation studies throughout an interdisciplinary approach [Amblard et al., 2015]. Linguistic analysis of dialogues between schizophrenics and psychologists has revealed specific language-driven manifestations of cognitive dysfunction. Therefore, our task should be both providing a way to model dialogue in a correct conversation setting and a way to handle, yet acknowledge, ongoing conversational disorders. Ultimately, answering these questions could lead to a major breakthrough in the field of diagnosis assistance.

This document presents the work that I conducted as part of my second year of Master's degree in Cognitive Sciences and Applications, specialized in Natural Language Processing, at the University of Lorraine. It was realized during a five-months internship at Loria (INRIA Nancy Grand Est) in the SÉMAGRAMME team, under the supervision of Maxime Amblard. The aim of this work is to come up with an adaptive logical model for a specific type of dialogical interactions in a dynamic framework, using and extending theory and methods developed by the SÉMAGRAMME team. It continues the previous Master thesis completed in the team [Tiv, 2016].
We begin by presenting the question of semantics of dialogue modeling in its scientific and socio-linguistic context. Then, we focus on the theoretical frameworks and methods used to address the problem. Starting by an introduction to TTDL, we continue by showing our first personal contributions by introducing a formalization of frame semantics. Section 4 presents and adapts the model from the previous Master thesis completed in the team. The following sections display our contributions to modifying and extending this model. Finally, we put our model in perspective, testing it on real-life data.

Contents

1 Motivation and state of the art

Natural language can be studied from different points of view, each focusing on specific features such as the sound of it (phonology), the look of it (morphology), its organization (syntax), its meaning (semantics), its meaning in context (pragmatics). In computational semantics, the main goal is to automate the process of reasoning on meaning as much as the articulation of meaning representations. Therefore, the first question that has to be solved concerns the representation of meaning in natural language – what formalization can and should be used?

1.1 Semantics of discourse

Richard Montague provided a possible answer, linking semantics, logic and language. It follows the modern formulation of the principle of compositionality by Gottlob Frege, stating that the meaning of a complex expression is determined by the meanings of its constitutive expressions and the rules used to combine them. Montague semantics [Montague, 1973] is based on Church's simple type theory [Church, 1940], giving the representation of constituents' meanings, which can then be combined according to syntax-driven rules in order to compute the representation of the complex expression meaning. Yet, Montague's approach only takes into consideration isolated sentences, thus mishandling long-scope linguistic phenomena that are specific to discourse, such as pronoun binding and anaphora – Montague semantics lacks dynamical aspects.

In order to attend to these issues, researchers from the field have developed multiple frameworks based on the ideas of Montague semantics but adding dynamicity to it. The first were Dicourse Representation Theory (DRT), introduced by Hans Kamp in [Kamp, 1981], and File Change Semantics, introduced by Irene Heim in [Heim, 1982]. Though those were developed independently, they share one main idea – consider meaning of segments composing a discourse as a relation between meanings of previous and yet-to-come sentences, as objects that can entail modifications in previously conducted analysis of discourse. The core notion underlying DRT's framework is the one of *discourse representation structures* (DRS) – a representation of discourse that is being enriched as new sentences are analyzed, as a human hearer would proceed with a mental representation of a discourse [Kamp and Reyle, 2013]. Unfortunately, though the idea of DRS is natural, its rigorous definition does not permit an easy merging of two DRSs. Indeed, as DRSs use sets of reference markers that should be understood both as existential quantifiers (necessary for the representation's sake) and free variables (needed to be able to enrich representations as the discourse continues), combining two DRSs in order to create one containing all the information provided by the previous two is a complex task that fails if variables assignments from both DRSs accidentally collide. Therefore, though DRT provides a way of dealing with dynamicity in discourse, it presents the great disadvantage of not being fully compositional.

Since the 90's, Nicolas Asher and Alex Lascarides started developing a theory that would extend dynamic semantics by exploiting additional information contained in discourse. On top of the order of sentences in a discourse, their inherent syntax and the way they can be composed, the idea underlying Segmented Discourse Representation Theory (SDRT) is to take into consideration rhetorical relations between segments of discourse [Asher and Lascarides, 2003]. SDRT is then defined as an extension of DRSs, which are enriched with rhetorical discourse relations such as `Explanation`, combined with semantics that is an extension of DRT's semantics enriched with the handling of rhetorical relations. This multi-layered structure allows to handle a wider collection of discourse-related phenomena. In particular, it copes with non-chronological accounts of events (ex: John smiled. Mary gave him flowers the day before.). Yet, the multiplication of embedded levels of modeling entails a complication of the resulting model of discourse, making it heavy to use.

Type Theoretical Dynamic Logic (TTDL) [de Groote, 2006], based on simply typed λ-calculus [Church, 1940], also solves dynamicity-related issues by combining Montague semantics with the idea of compositional meaning, depending on the surroundings of the considered piece of discourse. Discourse can then be seen as the combination of a context, storing known information (called left context) and a continuation in the computer science meaning of the word (called right context). Storing the linear "history" of discourse and providing operators to access this "history" allows to handle dynamical linguistic phenomena. The peculiarity of TTDL in the field of dynamic semantics lies in the idea of solving dynamical issues by defining the left context in a specific way. Therefore, TTDL constitutes a theory with a relatively light flat formalism (relying on λ-calculus only), accounting for dynamic issues of discourse and coping with compositionality (as it is compositional in its definition). See Section 2 for a detailed presentation of TTDL.

1.2 Semantics of dialogue

Now, if we consider dialogue as the combination of a context (previous utterances) and a continuation (utterances that yet have to come), it seems that it should be possible to obtain a dynamic semantics (here, TTDL) representation of a dialogical interaction. Yet, it appears that dialogue cannot be simply considered as a sub-type of general discourse, as dialogue involves more than one participant and therefore, discourse phenomena appear in an altered way in dialogue. Multiple participant nature of dialogue triggers issues linked to the notion of *common ground* as it is defined in [Stalnaker, 2002]:

> "It is common that ϕ in a group if all members *accept* (for the purpose of the conversation) that ϕ, and all *believe* that all accept that ϕ, and all believe that all believe that all accept that ϕ, etc."

It follows that it is absolutely not granted for a proposition that is uttered by one of the participants of the conversation to be common. It is usual for an utterance to first be discussed before being accepted. Consider the following dialogue between **A** and **B**:

Example 1 (A simple dialogue)**.**

A₁ *You turn left here.*

B₂ *There?*

A₃ *No, here.*

A first utters a declarative sentence. Then, **B** asks a question about one of the constituents of **A**'s utterance; thus, **B** does not directly accept **A**'s proposition. **A** then answers **B**'s question negatively, therefore switching its content to the opposite one: the utterance that might be further discussed is at this point "You turn left here, not there.".
Though this first example is quite small, it already shows that modeling dialogical interaction involves taking into consideration several complex phenomena that must be added to the ones encountered in discourse modeling.

Semantics of dialogue is a fundamental subject that has been extensively studied from numerous points of view. As written by Jonathan Ginzburg in [Ginzburg, 2016], the main considered topic is the context. It both governs the conversation, allowing or forbidding possible dialogical moves, and stores seeds of future dialogical opportunities. Ginzburg then highlights two major problems that semantic analysis in dialogue should take into consideration:

Conversational relevance: given that a conversation is in a certain state, what utterances can be produced coherently by each conversational participant in that state?

Conversational meaning: what conversational states are appropriate for a given word/construction and what import will that word have in such a state?

Conversation relevance is a notion that cannot be easily stated as a computational semantics problem. On the other hand, conversational meaning naturally falls into the scope of computational semantics and pragmatics. Yet, although meaning is a major topic of interest in natural language processing studies, viewing it as a dialogue issue implies taking into consideration the existence of linguistic structures specific to a conversational setting. It appears that dialogical studies so far have tried to focus on one core dialogical phenomenon at once, among which some of the most frequently found are non sentential utterances, described by Ludwig Wittgenstein in [Wittgenstein, 1953] and Why-Because Systems with Questions, introduced by Charles Leonard Hamblin in [Hamblin, 1970]. The latter presented a first model which attempted to describe a dialogical setting in terms of propositional logic predicates, each having the possibility to affect a set of commitments. This idea can be found nowadays in models such as those introduced in [Schlöder et al., 2016] but it is still limited to only cooperative dialogical settings, following the tradition set by [Grice, 1975], and dialogues presented in the SLAM corpus cannot be considered as fully cooperative.

The SLAM corpus is composed of 30 semi-structured interviews of either schizophrenics under medication, schizophrenics without medication or control patients (without any known diagnosis) by psychologists [Amblard et al., 2015]. It is natural to think that those dialogues would present an easily observable asymmetry in the participants' roles as the psychologists should lightly structure (control) the interviews while the patient's role would be a passive one; yet, the first observations of the corpus contradict this hypothesis. Some of the conversations are explicitly guided by the patient, who asks most of the questions; in some other, the patient is totally passive, compelling the psychologist to structure the interview with closed (*yes/no-*) questions only, as the patient answers solely by yes or no or by repeating the question in an assertive way, and doesn't elaborate any further-going sentences.
Then, would it be possible to consider modeling dialogues from SLAM as non-cooperative ones? Douglas Walton presents a first insight of one type of non-cooperative dialogue in [Walton, 2003] with a study, from the argumentation theory point of view, of interrogation. It appears that interrogations stand apart from cooperative types of dialogues in that they include large non-cooperative negotiation phases that cannot be modeled with the tools used for dialogues where both participants try to establish a well-going communication. Walton's work has been continued extensively as shown in [Plüss, 2014]. This thesis work focuses mainly on political interviews – dialogical settings in which the dialogical interaction does not fail at a psycho-linguistic level but where one of the participants punctually refuses to collaborate. Brian Plüss' introduces a game theory point of view in the field of dialogue modeling along with a way to quantify non-cooperativity in a dialogue. Yet, the subtle difficulty when considering dialogues from SLAM is that they cannot be classified as non-cooperative ones. Indeed, all the participants are willing to talk to the psychologist. The apparent non-cooperation in these interviews lies precisely at the psycho-linguistic level – conversation failures are inherent to the behaviour of the person being interviewed.

[Lecomte et al., 2007] and later [Fouqueré and Quatrini, 2012] introduce Ludics, a framework for dialogue modeling that is free from cooperation considerations, as it focuses solely on the logics side of argumentation dialogues. Modeling excerpts of SLAM with Ludics is an on-going process, yet, as Ludics ignores the linguistic side of dialogical interactions, it is not suitable for our purpose – it would only be able to detect a pure logical failure in a dialogue, not a semantic one.

The first necessary step for an attempt to model dialogues from the SLAM corpus is to get the ability to model conversational settings involving questions and answers. A starting point is provided in [Tiv, 2016] and is based on a combination of Type Theoretical Dynamic Logic and Frames Semantics. The latter allows to handle questions and answers by making their target explicit, while the first is responsible for the combination of utterances in the correct order. In the following, we will present both semantics with examples of modeling.

2 Type Theoretical Dynamic Logic

Though Montague semantics, presented above, offers numerous expressivity advantages due to the possibility to use Frege's principle of compositionality, it lacks dynamical notions. The idea underlying the work presented in [de Groote, 2006] is to show that this weakness can be fixed using simply typed λ-calculus to represent meaning and β-reduction to introduce dynamicity. To do so, a parallel is drawn between the computation of the representation of a piece of discourse and the execution of a program by adapting the notion of *continuation* – a programming environment/context within which some code is executed – to natural language. Thus, the continuation of a piece of discourse corresponds to the bits coming right after the first one and relying on the first one (that can be empty) to be understood.
The following presents TTDL as introduced in [de Groote, 2006].

2.1 Formal set-up

First, we consider three atomic types. Two are directly inherited from Church's simple type theory and the third one is specifically introduced for the purpose of TTDL.

Definition 1 (Basic types)**.**

- ι : *individal/entity*
- o : *proposition*
- γ : *left context*

Then, a sentence is represented as a combination of its left and right contexts. As a sentence is ultimately represented as a proposition, it should be of type o. Therefore, the type of the right context is $\gamma \to o$, which is consistent with the denotational semantics idea of continuation introduced above.

$$\underbrace{\underbrace{\overbrace{\rule{4cm}{2pt}}^{\textit{left context}}}_{\gamma}\ \bullet\ \underbrace{\overbrace{\rule{4cm}{2pt}}^{\textit{right context}}}_{\gamma \to o}}_{o}$$

As discourse is here considered as ultimately constructed in the same way as a single sentence, we have:

Proposition 1. *Let s be the syntactic category of sentences, d be the syntactic category of discourses. Then, the corresponding semantic interpretations are:*

$$[\![s]\!] = \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

$$[\![d]\!] = \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$$

It is important to notice that this proposition does not give a constructive way to compute the representation of the meaning of discourse by composing meanings of its constituent sentences. To address that issue, we define the linear aggregation operator . as follows:

Definition 2 (Linear aggregation operator).
Let D be a piece of discourse, S be a sentence. Then, the semantics of the discourse composed of D followed by S is given by:

$$[\![\mathsf{D.S}]\!] \ := \ \lambda e \phi.\, [\![\mathsf{D}]\!]\, e \ (\lambda e'.\, [\![\mathsf{S}]\!] e' \, \phi)$$

$$\overbrace{\qquad\qquad\qquad\qquad}^{e:\gamma}\ \mathsf{D}.\underbrace{\mathsf{S}\ \overbrace{\qquad\qquad\qquad\qquad}^{\phi:\gamma\to o}}_{\lambda e'.[\![\mathsf{S}]\!]\, e'\, \phi\, :\gamma\to o}$$

The term takes two arguments: e of type γ, left context of $(D.S)$, and ϕ of type $\gamma \rightarrow o$, right context of $(D.S)$. The result is then composed of the representation of D combined with e (left context of $(D.S)$) and a third term. This third term corresponds to the newly computed continuation of D, as it takes as argument e', the left context of S (combination of e and the content of D) and returns a result composed of the representation of S combined with ϕ, right context of $(D.S)$ and thus of S.

Yet, the linear aggregation operator alone does not give a solution to handle operators in context and solving anaphora. To address these issues, it is necessary to define objects that will allow us to dynamically update and access the context:

Definition 3 (Context aggregation operator and oracles).
Let c be a context (of type γ). We hereafter consider that c can be viewed as a finite set of individuals. The context aggregation operator*, necessary to add new individuals to the context set, is typed as follows:*

$$\_ :: \_ : \iota \rightarrow \gamma \rightarrow \gamma$$

$\_ :: \_$ *takes two arguments: a new individual of type ι and c. It results in the updated context. Conversely, if x is a pronoun, we type the corresponding* oracle *(a choice operator retrieving the right individual for the anaphora solving from the context) as follows:*

$$sel_x : \gamma \rightarrow \iota$$

sel_x takes c and retrieves from it an individual corresponding to the pronoun x. For example, if $c = (Mary \ :: \ John)$, $sel_{her}(c) = Mary$.

The following work is presented under the hypothesis of the existence and good behaviour of these oracles. We do not discuss here the technical issues of anaphora resolution in a real-life setting.

2.2 Example

Now, let us consider the following toy example of discourse:

Example 2 (A small example of discourse)**.**

> *(2) John loves Mary. He smiles at her.*

In order to compute its semantic representation, we denote by D the first sentence and by S the second:

$$D = \text{John loves Mary.}$$

$$S = \text{He smiles at her.}$$

Then, we compute $[\![D.S]\!]$ following the previously given definition of the linear aggregation operator (def. 2). To properly conduct this computation, we follow the syntactical structure of the sentences.

2.2.1 Constituent analysis

First, we need to give the syntactic categories of the lexical constituents of D and S. After that, we define the types of corresponding λ-terms, which helps us to give the proper semantic representations of the constituents of D and S.

Syntactic categories

$$\begin{aligned} \text{John, Mary, he, her} &: NP \\ \text{loves, smiles\_at} &: TV = NP \to NP \to S, \end{aligned}$$

Where NP stands for the syntactic category of noun phrases, TV the syntactic category of transitive verbs and S the syntactic category of sentences.

Typing

$$\begin{aligned} [\![s]\!] &= \gamma \to (\gamma \to o) \to o \\ [\![np]\!] &= (\iota \to [\![s]\!]) \to [\![s]\!] \\ [\![tv]\!] &= [\![np \to np \to s]\!] \\ &= [\![np]\!] \to [\![np]\!] \to [\![s]\!] \end{aligned}$$

Where $[\![s]\!]$ is the type of sentences, $[\![np]\!]$ the type of nous phrases and $[\![tv]\!]$ the type of transitive verbs.

Semantic representation

$$\begin{aligned} [\![Mary]\!] &= \lambda\psi e.\psi\ \mathbf{m}(\mathbf{m} :: e) \\ [\![John]\!] &= \lambda\psi e.\psi\ \mathbf{j}(\mathbf{j} :: e) \\ [\![he]\!] &= \lambda\psi e\phi.\psi(sel_{he}e)e\phi \\ [\![her]\!] &= [\![she]\!] \\ &= \lambda\psi e\phi.\psi(sel_{she}e)e\phi \end{aligned}$$

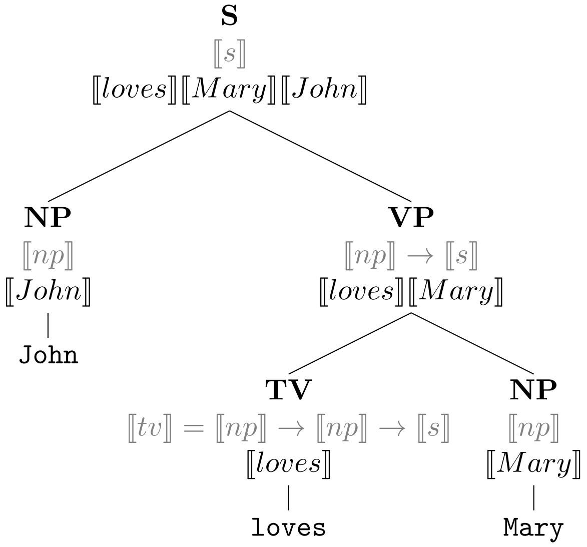
The semantic representation of “her” and “she” coincide in this case because of the chosen example; in example 2, “her” refers to “Mary” and not “Mary’s”.

$$\begin{aligned} \llbracket loves \rrbracket &= \lambda o s e \phi . s(\lambda x e . o(\lambda y e . \mathbf{love}\ x\ y \wedge \phi e) e) e \\ \llbracket smiles\_at \rrbracket &= \lambda o s e \phi . s(\lambda x e . o(\lambda y e . \mathbf{smile}\ x\ y \wedge \phi e) e) e \end{aligned}$$

Where o and s refer, respectively, to the object and the subject of the transitive verbs.

2.2.2 Computation

First, we compute the semantic representation of D, according to the syntactic structure of the sentence.



Semantic representations of the intermediary nodes of the tree are computed bottom-up. According to the types of $\llbracket tv \rrbracket$ and $\llbracket np \rrbracket$, $\llbracket loves \rrbracket$ is the functor applied to the argument $\llbracket Mary \rrbracket$. Similarly, typing gives us that $\llbracket loves \rrbracket \llbracket Mary \rrbracket$ is the functor applied to the argument $\llbracket John \rrbracket$.

Unfolding the term by using β-reduction, we obtain the following detailed computation:

$$\begin{aligned}
\llbracket loves \rrbracket \llbracket Mary \rrbracket \llbracket John \rrbracket &= \Big(\lambda ose\phi.s(\lambda xe.o(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)e\Big)\llbracket Mary \rrbracket \llbracket John \rrbracket \\
&\to_\beta \Big(\lambda se\phi.s(\lambda xe.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)e\Big)\llbracket John \rrbracket \\
&\to_\beta \Big(\lambda e\phi.\llbracket John \rrbracket(\lambda xe.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)e\Big) \\
&= \Big(\lambda e\phi.(\lambda\psi e.\psi\mathbf{j}(\mathbf{j} :: e))(\lambda xe.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)e\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda e.(\lambda xe.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)\mathbf{j}(\mathbf{j} :: e))e\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda xe.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ xy \wedge \phi e)e)\mathbf{j}(\mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda e.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ \mathbf{j}y \wedge \phi e)e)(\mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.\llbracket Mary \rrbracket(\lambda ye.\mathbf{love}\ \mathbf{j}y \wedge \phi e)(\mathbf{j} :: e)\Big) \\
&= \Big(\lambda e\phi.(\lambda\psi e.\psi\mathbf{m}(\mathbf{m} :: e))(\lambda ye.\mathbf{love}\ \mathbf{j}y \wedge \phi e)(\mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda e.(\lambda ye.\mathbf{love}\ \mathbf{j}y \wedge \phi e)\mathbf{m}(\mathbf{m} :: e))(\mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda ye.\mathbf{love}\ \mathbf{j}y \wedge \phi e)\mathbf{m}(\mathbf{m} :: \mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.(\lambda e.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \phi e)(\mathbf{m} :: \mathbf{j} :: e)\Big) \\
&\to_\beta \Big(\lambda e\phi.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)\Big)
\end{aligned}$$

Thus,

$$\llbracket D \rrbracket = \lambda e\phi.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)$$

Similarly, the computation of $\llbracket S \rrbracket$ gives us:

$$\llbracket S \rrbracket = \lambda e\phi.\mathbf{smile}\ (sel_{he}e)(sel_{she}e) \wedge \phi e$$

Now, we can compute the semantics of $\llbracket D.S \rrbracket$:

$$\begin{aligned}
\llbracket D.S \rrbracket &= \lambda e\phi.\llbracket D \rrbracket e\Big(\lambda e'.\llbracket S \rrbracket e'\phi\Big) \\
&= \lambda e\phi.\Big(\lambda e\phi.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)\Big)e\Big(\lambda e'.\llbracket S \rrbracket e'\phi\Big) \\
&\to_\beta \lambda e\phi.\Big(\lambda\phi.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)\Big)\Big(\lambda e'.\llbracket S \rrbracket e'\phi\Big) \\
&\to_\beta \lambda e\phi.\Big(\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge (\lambda e'.\llbracket S \rrbracket e'\phi)(\mathbf{m} :: \mathbf{j} :: e)\Big) \\
&\to_\beta \lambda e\phi.\Big(\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \llbracket S \rrbracket(\mathbf{m} :: \mathbf{j} :: e)\phi\Big) \\
&= \lambda e\phi.\Big(\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge (\lambda e\phi.\mathbf{smile}\ (sel_{he}e)(sel_{she}e) \wedge \phi e)(\mathbf{m} :: \mathbf{j} :: e)\phi\Big) \\
&\to_\beta \lambda e\phi.\Big(\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \Big(\lambda\phi.\mathbf{smile}\ (sel_{he}(\mathbf{m} :: \mathbf{j} :: e))(sel_{she}(\mathbf{m} :: \mathbf{j} :: e)) \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)\Big)\phi\Big) \\
&\to_\beta \lambda e\phi.\Big(\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \mathbf{smile}\ (sel_{he}(\mathbf{m} :: \mathbf{j} :: e))(sel_{she}(\mathbf{m} :: \mathbf{j} :: e)) \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)\Big)
\end{aligned}$$

Unfolding the selection operator, we have:

$$(sel_{he}(\mathbf{m} :: \mathbf{j} :: e)) = \mathbf{j}$$
$$(sel_{she}(\mathbf{m} :: \mathbf{j} :: e)) = \mathbf{m}$$

Which gives us the desired final representation of D.S

$$[\![D.S]\!] = \lambda e \phi.\mathbf{love}\ \mathbf{j}\ \mathbf{m} \wedge \mathbf{smile}\ \mathbf{j}\ \mathbf{m} \wedge \phi(\mathbf{m} :: \mathbf{j} :: e)$$

that matches the intuition of this discourse's elements structure. This semantic representation displays all the advantages of the simplicity of Montague's approach combined with the newly possible handling of dynamic aspects. As shown in this section, it is now possible to solve anaphoras by considering discourse in a dynamic way. Similarly, it is possible to handle dynamic quantification. Yet, adapting TTDL to dialogue modeling is not straight-forward. Consider a dialogue involving two participants **A** and **B**. If **A** utters something, we should add that utterance to what we will consider to be the dialogical context. Yet, if **B** disagrees with **A**'s proposition, **B** will argue against it, and might convince **A**, changing **A**'s point of view. After that, **A** might utter a proposition which will be in contradiction with its previous one, introducing an internal non-coherence in the dialogical context. Thus, it is necessary to introduce an intermediary representation mean which will handle the negotiation phases of the dialogues and only store propositions in the common context after an agreement has been reached.

3 Frame semantics

The idea lying at the origin of frame semantics is very intuitive – a *frame* should be a representation of a situation, its participants, the semantic roles of those participants, and the relation between the participants and the situation. A frame as intended by Charles Fillmore in [Fillmore, 1982] is a cognitive semantic unit of information. Yet, formalizing this idea presents technical difficulties and can be done in many different ways; here, we choose to introduce a formal definition of semantic frames as typed base-labelled feature structure, following the construction exposed in [Kallmeyer and Osswald, 2014] and modifying it in perspective of our future applications.

3.1 Basic definitions

Frames are considered as typed base-labelled feature structures with relations that are built on top of a signature. First, we define the components of this signature and a some useful notation.

Definition 4 (Signature)**.**
A signature is a 3*-tuple* $\langle A, T, B\rangle$ *where:*

- A *is a finite set of* attributes *(also called* features*.*
- T *is a finite set of* types*.*
- B *is a finite set of* base labels*. Without loss of generality, we can assume that* $B = \{\boxed{0}, \boxed{1}, \boxed{2}, \ldots, \boxed{k}\}$*, where* $k \in \mathbb{N}$*.*

Keeping in mind our linguistic application of frames, we can consider the following example of a signature:

Example 3.
Let $\langle A, T, B\rangle$ be the signature such that:

- *A is a finite set of semantic role labels, such as* `Agent`*.*
- *T is a finite set of types, organized in an ontology. For example, if* `RUN` *is a type, it is a sub-type of the type* `ACTION`*.*
- *B is a finite set of arbitrarily ordered words. For example, if the aim is to represent a sentence, the first word of the sentence can be assigned the number* 0*.*

Next, we define frames. Here, the frames are introduced as typed base-labelled feature structures.

Definition 5 (Typed base-labelled feature structures (i.e frames))**.**
A typed base-labelled feature structure over the signature $\langle A, T, B\rangle$ is defined as a 5-tuple $\langle V, \delta, \delta_{neg}, \tau, \beta\rangle$ such that:

- *V is a finite set of* nodes*.*
- *δ is a partial function from $V \times A$ to V called the* transition function*.*
- *δ_{neg} is a partial function from $V \times A$ to V called the* negation function*, such that, for all $v \in V, a \in A$, if $\delta(v, a)$ is defined, then $\delta_{neg}(v, a)$ is not. We define $\hat{\delta}$ as an extension of δ and δ_{neg}. $\hat{\delta}$ is a partial function from $V \times A^+$ to V such that, if $v \in V$ and $p \in A^+$, $p = a_1, \ldots, a_n$[1]:*

$$\hat{\delta}(v,p) = \begin{cases} \delta(v,p) & \text{if } p = a_1 \in A \text{ and } \delta(v,p) \text{ is defined;} \\ \delta_{neg}(v,p) & \text{if } p = a_1 \in A \text{ and } \delta_{neg}(v,p) \text{ is defined;} \\ \hat{\delta}(\delta(v,a_1), a_2, \ldots, a_n) & \text{if } \delta(v,a_1) \text{ is defined;} \\ \hat{\delta}(\delta_{neg}(v,a_1), a_2, \ldots, a_n) & \text{if } \delta_{neg}(v,a_1) \text{ is defined.} \end{cases}$$

- *τ is a function from V to $\mathscr{P}(T)$ named* typing function[2]*.*
- *β is a partial function from B to V called the* base-labelling function*, such that*

$$\forall v \in V, \exists v' \in \beta(B) \text{ and an attribute path } p \in A^+ \text{such that } v = \hat{\delta}(v', p)$$

β is defined in such a way that every node is reachable from some base node, i.e from some element of $\beta(B) \subseteq V$, via attribute path transitions.
Now that the frame data structure is defined, let us consider the following utterance:

> $\mathbf{A_1}$ *You turn left here, not there.*

and propose a frame representation for it.

Example 4 (Frame representation)**.**
We consider the signature $\langle A, T, B\rangle$ over which we build the frame $\langle V, \delta, \delta_{neg}, \tau, \beta\rangle$, where:

- *A is the set of semantic roles labels (*`Agent`*,* `Location`*,* `Direction`*).*
- *T is the ontologically organized set of types (here, containing the type* `TURN`*).*

[1] Where A^+ denotes the set of non-empty strings of elements of A

[2] Where $\mathscr{P}(T)$ is the powerset of T.

- *B is the bag of words corresponding to the utterance: $\{you, turn, left, here, there\}$. It doesn't contain "not", interpreted as a logical operation.*
- *V contains five nodes (one per word in the bag of words).*
- *δ is represented in Figure 1 in full line.*
- *δ_{neg} is represented in Figure 1 in dashed line.*
- *τ assigns the type {TURN} to the node whose base label is* turn *and $\emptyset$ to the other nodes.*
- *β is represented in Figure 1 by the rectangle boxed nodes-round vertices pairs.*

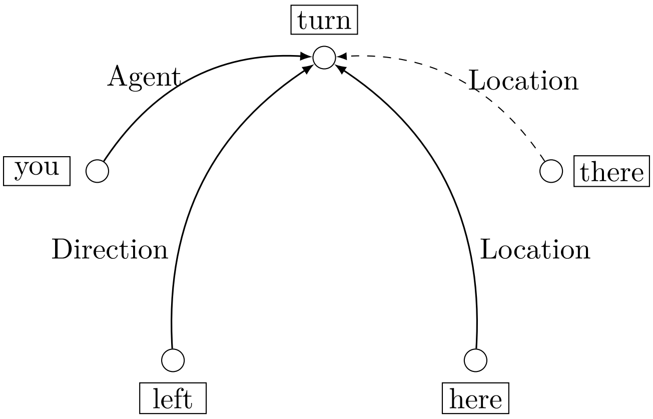


Figure 1: Possible graphical representation of a frame.

This formalization may seem unnecessary heavy, especially after comparing a frame representation of a sentence with a TTDL one. Yet, this rigorously defined mathematical structure allows to prove some properties that are necessary to compute a dialogue negotiation phase.

3.2 Formal approach

Drifting away from our linguistic application, we will now consider frames as the abstract objects defined above and present and explore few of their mathematical properties. First, we define a subsumption relation, introducing an ordering on frames.

Definition 6 (Subsumption).
Let $F_1 = \langle V_1, \delta_1, \delta^1_{neg}, \tau_1, \beta_1 \rangle$ and $F_2 = \langle V_2, \delta_2, \delta^2_{neg}, \tau_2, \beta_2 \rangle$ be two frames over $\langle A, T, B \rangle$. F_1 subsumes F_2 (denoted $F_1 \sqsubseteq F_2$) if there exists a morphism h from V_1 to V_2 such that:

- *if $\delta_1(v, f)$ is defined for $v \in V_1$ and $f \in A$, then $\delta_2(h(v), f) = h(\delta_1(v, f))$.*
- *if $\delta^1_{neg}(v, f)$ is defined for $v \in V_1$ and $f \in A$, then $\delta^2_{neg}(h(v), f) = h(\delta^1_{neg}(v, f))$.*
- *$\forall v \in V_1$, $\tau_1(v) \subseteq \tau_2(h(v))$.*
- *If $\beta_1(b)$ is defined for $b \in B$, then $h(\beta_1(b)) = \beta_2(b)$.*

It follows in particular that adding additional base labels to a frame (extending the set B) gives a more specific frame with respect to subsumption (see Figure 2).

We need to give an additional characterization of frames. We define an equivalence relation on frames, on top of which we can then define minimal frames with respect to a set of frames.

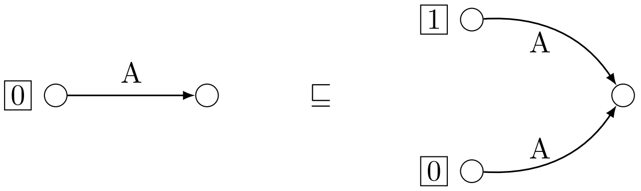


Figure 2: Strict subsumption between two frames.

Definition 7 (Equivalent frames, minimal frame)**.**
Let F_1 and F_2 be two frames. F_1 is equivalent *to F_2 (denoted $F_1 \cong F_2$) if and only if $F_1 \sqsubseteq F_2$ and $F_2 \sqsubseteq F_1$.*
Let $\mathscr{F}$ be a set of frames and let F be a frame, $F \in \mathscr{F}$. F is a minimal *frame of $\mathscr{F}$ if $\forall G \in \mathscr{F}, G \sqsubseteq F$ implies $G \cong F$.*

Now, we can define the union of two frames.

Definition 8 (Union)**.**
Let F_1 and F_2 be two frames, $\mathscr{F}_1 = \{F|F_1 \sqsubseteq F\}$ and $\mathscr{F}_2 = \{F|F_2 \sqsubseteq F\}$ be the sets of frames subsuming F_1 and F_2 respectively. Let $\mathscr{I} = \mathscr{F}_1 \cap \mathscr{F}_2 = \{F|(F_1 \sqsubseteq F) \wedge (F_2 \sqsubseteq F)\}$. Then, three cases are possible:

1. *$\mathscr{I}$ does not contain any minimal frame.*
2. *$\mathscr{I}$ contains at least two non-equivalent minimal frames.*
3. *$\mathscr{I}$ contains at least one minimal frame, and all its minimal frames are equivalent.*

In the first and the second cases, the union is not defined. In the third case, let $I \in \mathscr{I}$ be a minimal frame. We denote the union of F_1 and F_2 as $F_1 \sqcup F_2$ and we have $F_1 \sqcup F_2 \cong I$.

We want to be able to use and combine the frames, in order to get back to our linguistic applications. For now, the most frequent case of application which we will consider concerns two frames, one of which is empty. It corresponds, linguistically, to a situation where no previous context is available – for example, at the beginning of a conversation.
First, we give a formal definition of the empty frame:

Definition 9 (Empty frame)**.**
A frame is said to be empty *and is denoted by $[\ ]$ if is defined as a typed base-labelled feature structure $\langle V, \delta, \delta_{neg}, \tau, \beta\rangle$ over any signature $\langle A, T, B\rangle$ such that:*

$$V = \emptyset.$$

The proposition below subsequently follows:

Proposition 2.
Let F be a frame and $[\ ]$ be the empty frame. Then $F \sqcup [\ ] \cong F$.

Proof. Let $\mathscr{I}$ be the set $\{A|F \sqsubseteq A \wedge [\ ] \sqsubseteq A\}$. Then, by the definition of the empty frame, $[\ ] \sqsubseteq A$ for every frame A, which gives $\mathscr{I} = \{A|F \sqsubseteq A\}$. Let $G \in \mathscr{I}$ be a frame such that $G \sqsubseteq F$. Then, by the definition of $\mathscr{I}$, $F \sqsubseteq G$. Therefore, $F \cong G$. By the definition of a minimal frame, F is thus minimal, which means that $\mathscr{I}$ contains at least one minimal frame.
Let B be a minimal frame of $\mathscr{I}$. As $B \in \mathscr{I}$, $A \sqsubseteq B$ by the definition of $\mathscr{I}$. As B is a minimal frame of $\mathscr{I}$ and $A \in \mathscr{I}$, we have $B \sqsubseteq A$. Therefore, all minimal frames of $\mathscr{I}$ are equivalent. By the definition of the union, we have $F \sqcup [\ ] \cong F$. □

After exploring further the mathematical theory of frames, we are now able to use the frames to convey meaning representation.

4 Modeling – a first approach

The approach presented bellow, introduced in [Tiv, 2016], focuses on modeling propositions, questions and answers to set the common ground between dialogue participants. To this end, type theoretical continuation semantics is used together with frame semantics. A dialogue structure is then modeled with types and operators for utterances, while the content is represented with frames. Questions and answers are modeled with the notion of focus, symbolizing a request about a specific argument. As a result, a dialogue is seen as a sequence of typed utterances, and its meaning as frames, combined by means of continuation semantics. The following presents the construction introduced in [Tiv, 2016], enriched in light of the formalization of the previous section.

4.1 Dialogical settings

For the sake of simplicity, though we will keep in mind the formal definition of frames given above, we will use a simplified representation in what follows. As an illustration, the frame representation of $\mathbf{A_1}$ "You turn left here." is hereafter given by:

$$[\![A_1]\!] = \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } here \end{bmatrix}$$

Then, $\mathbf{B_2}$ "There?" is a question asked by **B**. It relies on the previous utterance $\mathbf{A_1}$ as it queries one of its elements – the base label corresponding to the feature `Location`. $\mathbf{B_2}$ is represented by a *pending* frame, as it is dependent on the context and waits for a second argument (an answer) to be solved, which here means to compute the representation of the dialogue. This is what a possible representation for $\mathbf{B_2}$ looks like:

$$[\![B_2]\!] = \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } ?(there) \end{bmatrix}$$

In order to represent the focus of the questions and the answers, a λ-abstraction is introduced. Here, the queried feature is `Location`.

$$\lambda l. \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } l \end{bmatrix}$$

To solve the questions, it is necessary to define an operator that is able to retrieve the base label corresponding to the interrogated feature as well as the λ-abstraction representing the modification path inside the frame.

Definition 10 (Find operator)**.**
Let v be a feature. We hereafter consider that frames (except pending ones) are of type γ. The corresponding $find_v$ operator is typed as follows:

$$find_v : \gamma \to v \times (v \to \gamma)$$

Example 5. *Applying the operator $find_{Loc}$ to the frame representation of A_1, we obtain:*

$$(here, \lambda l. \begin{bmatrix} TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } l \end{bmatrix})$$

As mentioned above, this model focuses on dialogical settings that involve only three types of dialogical acts: assertions, questions and answers. Moreover, we consider only questions concerning a feature v and corresponding answers (about a feature v). We therefore define:

Definition 11. *Let v be a feature. Let u be the syntactic category of assertions, q_v be the syntactic category of questions concerning v, a_v be the syntactic category of answers concerning v. Then, the corresponding semantic interpretations are:*

$$\begin{aligned} [\![u]\!] &= \gamma \rightarrow \gamma \\ [\![q_v]\!] &= \gamma \rightarrow v \times (v \rightarrow \gamma) \\ [\![a_v]\!] &= v \times (v \rightarrow \gamma) \rightarrow \gamma \end{aligned}$$

Then, we can present a constructive way to combine the utterances in order to compute the representation of the meaning of the dialogue. To do so, we define three linear aggregation operators $.^u$, $.^q$ and $.^a$ as follows:

Definition 12.

$$\begin{cases} .^u : u \rightarrow u \rightarrow u \\ .^u = \lambda U_1 U_2 c. U_2(U_1 c) \end{cases}$$

$$\begin{cases} .^q : u \rightarrow q_v \rightarrow q_v \\ .^q = \lambda U Q c. Q(U c) \end{cases}$$

$$\begin{cases} .^a : q_v \rightarrow a_v \rightarrow u \\ .^a = \lambda Q A c. A(Q c) \end{cases}$$

The operator $.^u$ takes two assertions to produce a third one, combination of the first two with the context. The operator $.^q$ takes an assertion and a question to produce a question that is defined on a specific feature of the assertion and takes the context into consideration. The operator $.^a$ takes a question and an answer to produce an utterance, solution of the question. Setting the types of these operators allows to control the well behaviour of the dialogical process – a question can follow a proposition, an answer can follow a question, but an answer cannot follow a proposition. We will see examples of these dialogical acts combinations bellow.

4.2 Example

The following presents the full computation of a semantic representation of a dialogue excerpt, using the previously developed model. Let us come back to the previously considered dialogue example:

Example 6 (A simple dialogue).

$\mathbf{A_1}$ *You turn left here.*

$\mathbf{B_2}$ *There?*

$\mathbf{A_3}$ *No, here.*

To compute its semantic representation, we need to identify the nature of its constitutive dialogical acts. This identification process can be automatized using machine learning techniques, as presented in [Vosoughi and Roy, 2016].
In Example 6, $\mathbf{A_1}$ is an assertion, $\mathbf{B_2}$ a question about the feature `Location` of $\mathbf{A_1}$. $\mathbf{A_3}$ is an answer to $\mathbf{B_2}$. Therefore, the semantic representation of Example 6 is the semantic representation of $A_1.^q B_2.^a A_3$. To proceed with the computation, we need to define the λ-frame-terms corresponding to each utterance.

4.2.1 Constituents analysis

Types

$$\begin{aligned} [\![A_1]\!] &: [\![u]\!] \\ [\![B_2]\!] &: [\![q_{Loc}]\!] \\ [\![A_3]\!] &: [\![a_{Loc}]\!] \end{aligned}$$

Representations

$$[\![A_1]\!] = \lambda c. \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } here \end{bmatrix} \sqcup c$$

$$\begin{aligned} [\![B_2]\!] &= \lambda e.\texttt{let}(o,c) = find_{Loc}(e) \texttt{ in } (there, c) \\ &\equiv \lambda e.\Big((\lambda(o,c).(here, c))\,(find_{Loc}(e)) \Big) \\ &\equiv \lambda e.\Big(there, \pi_2(find_{Loc}(e)) \Big) \end{aligned}$$

$$[\![A_3]\!] = \lambda(x,e).e(here)$$

4.2.2 Computation

First, we compute the combination of $\mathbf{A_1}$ with $\mathbf{B_1}$. As $\mathbf{A_1}$ is an assertion and $\mathbf{B_2}$ is a question, we combine them using the operator $.^q$:

$$\begin{aligned}
[\![A_1.^q B_2]\!] &= \Big(\lambda UQ\ c.Q(U\ c)\Big)[\![A_1]\!][\![B_2]\!] \\
&\rightarrow_\beta \Big(\lambda Q\ c.Q([\![A_1]\!]\ c)\Big)[\![B_2]\!] \\
&\rightarrow_\beta \lambda c.[\![B_2]\!]\Big([\![A_1]\!]\ c\Big)
\end{aligned}$$

Then, we can aggregate $\mathbf{A_3}$ to the result, using the operator $.^a$, as $\mathbf{A_3}$ is an answer.

$$\begin{aligned}
[\![A_1.^q B_2]\!].^a[\![A_3]\!] &= [\![A_1.^q B_2.^a A_3]\!] \\
&= \Big(\lambda QA\ c.A(Q\ c)\Big)[\![A_1.^q B_2]\!][\![A_3]\!] \\
&\rightarrow_\beta \Big(\lambda A\ c.A([\![A_1.^q B_2]\!]\ c)\Big)[\![A_3]\!] \\
&\rightarrow_\beta \lambda c.[\![A_3]\!]\Big([\![A_1.^q B_2]\!]\ c\Big) \\
&= \lambda c.[\![A_3]\!]\Big((\lambda c.[\![B_2]\!]([\![A_1]\!]\ c))\ c\Big) \\
&\rightarrow_\beta \lambda c.[\![A_3]\!]\Big([\![B_2]\!]([\![A_1]\!]\ c)\Big)
\end{aligned}$$

Finally, as we need to take the context of the dialogue into consideration but we consider this dialogue alone, we can apply the previous result to the empty context.

$$\begin{aligned}
[\![A_1.^q B_2.^a A_3]\!]c_e &= \lambda c.[\![A_3]\!]\Big([\![B_2]\!]([\![A_1]\!]\ c)\Big)c_e \\
&\rightarrow_\beta [\![A_3]\!]\Big([\![B_2]\!]([\![A_1]\!]\ c_e)\Big)
\end{aligned}$$

Now, we can unfold the computation with explicit frames representations. We consider that the semantic representation of the empty context c_e is an empty frame. First, we compute:

$$\llbracket A_1 \rrbracket c_e = (\lambda c. \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ \mathit{here} \end{bmatrix} \sqcup c) c_e$$

$$\rightarrow_{\beta} \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ \mathit{here} \end{bmatrix} \sqcup c_e$$

$$= \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ \mathit{here} \end{bmatrix} \sqcup [\,]$$

$$\cong \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ \mathit{here} \end{bmatrix}$$

$$= \textcircled{1}$$

This computation involves using the $\sqcup$ operator defined above.
Then, we apply $\llbracket B_2 \rrbracket$ to the result of the previous computation:

$$
\begin{aligned}
[\![B_2]\!]\ \textcircled{1} &= \lambda e.(there, \pi_2(find_{Loc} e))\ \textcircled{1} \\
&= \lambda e.(there, \pi_2(find_{Loc} e)) \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } here \end{bmatrix} \\
&\to_\beta\ (there, \pi_2(find_{Loc} \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } here \end{bmatrix})) \\
&= (there, \pi_2(here, \lambda l \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } l \end{bmatrix})) \\
&= (there, \lambda l. \begin{bmatrix} \quad TURN \\ \texttt{Ag: } B \\ \texttt{Dir: } left \\ \texttt{Loc: } here \end{bmatrix}) \\
&= \textcircled{2}
\end{aligned}
$$

Finally, we apply $[\![A_3]\!]$ to what preceded and we obtain:

$$\begin{aligned}
[\![A_3]\!]\ \textcircled{2} &= (\lambda(x,e).e\ \mathit{here})\ \textcircled{2} \\
&= (\lambda(x,e).e\ \mathit{here})(\mathit{there}, \lambda l. \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ l \end{bmatrix}) \\
&\to_\beta (\lambda l. \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ l \end{bmatrix})(\mathit{here}) \\
&= \begin{bmatrix} \mathit{TURN} \\ \texttt{Ag:}\ B \\ \texttt{Dir:}\ \mathit{left} \\ \texttt{Loc:}\ \mathit{here} \end{bmatrix}
\end{aligned}$$

Which corresponds to the intuition of this dialogue's result.

We built our model as an application of the Type Theoretical Dynamic Logic framework to a semantic based on frames. This construction allowed us to access a way of computing an intermediary representation of matter under discussion during a negotiation phase of a dialogue. The following sections go into the handling of those resulting intermediary representations.

5 Common (dialogical) context

Constructing a model based on TTDL allows us to use the idea of storing information in a structure that we call context and at the same time allowing following discourse (continuation) to access it. Yet, as mentioned above, transition from a general discourse framework to a dialogue one does not go so smoothly concerning the context, as we need to add control in the storage operations, because of the negotiation phases that can change the content of what will be stored. Considered dialogues as an aggregation of negotiation phases starting with an utterance and finishing with an answer and providing a way to conduct computations of frames has allowed us to define the model introduced in Section 4. The aim of the following section is to discuss and define context storage operations and their usages.

5.1 Operations on context

To build our "dialogical context" we follow the same intuition and formalization as for the context in TTDL – an aggregation of representations of previous pieces of discourse, of type γ – yet, following the definition of the common ground given by Stalnaker (see Section 1), we want to control what information is being stored in the dialogical context.
First, we need an operation that will allow us to store a frame in the context once the participants of the dialogue have reached an agreement at the end of the negotiation phase.

Example 7 (Reaching an agreement)**.**

$\mathbf{A_1}$ *Is this chair new?*

$\mathbf{B_2}$ *Yes, it is.*

We will hereafter perform this operation through the use of a *storing* operator. Yet, sometimes, the negotiation phase ends with a disagreement. Though a further discussion should be conducted, we hereafter, for the sake of simplicity, use the *dismiss* operator in such cases. As its name suggests, the dismiss operator does not store the frame resulting from the negotiation and simply shifts the computation of the context on to the next negotiation phrase of the dialogue.

Example 8 (Disagreement in dialogue)**.**

$\mathbf{A_1}$ *Should we buy a new chair?*

$\mathbf{B_2}$ *No, our old ones are still good!*

$\mathbf{A_3}$ *Yet, I think we should.*

In this case, if the dialogue stops there, no agreement has been reached. Therefore, the resulting frame has been dismissed.
In the following, we will discuss the different configurations of negotiation phases and conclude to which operator should be used in each case.

5.2 Closed questions

A closed question (also called *yes/no*-question) is used to ask a question with yes or no as an answer [Aarts et al., 2014]. According to [Ginzburg and Sag, 2000], in a real-life dialogical setting, it is important to distinguish two parts in an answer to a closed question. Consider the following example:

Example 9.

$\mathbf{A_1}$ *Do you live in Paris?*

$\mathbf{B_2}$ *Yes, near the Louvre.*

Indeed, the answer begins, as expected, with “Yes”, called hereafter the *short answer* (following the terminology of [Ginzburg and Sag, 2000]). Yet, the answer does not stop there and continues; this part corresponds to what is called *aboutness* – it adds precisions to $\mathbf{A_1}$.

Against what could be expected from their name, *yes/no*-questions can be answered by three types of short answers: “yes”, “no”, but also “maybe” (as well as “probably”, “perhaps”, etc.). When the short answer is “yes”, it triggers the action of the *storage* operator for the approved frame. It is important to note here that if the short answer “yes” is followed by an aboutness part, then the frame being stored is the one containing the information added in the aboutness. When the short answer is “no”, it would be natural to think that, as no agreement has been reached, the frame under negotiation should be dismissed. Yet, this would actually lead to a loss of information. Indeed, consider the following example:

Example 10 (No)**.**

A$_1$ *Do you live in Paris?*

B$_2$ *No.*

This negotiation phase gave us the information that the content of the feature `Location` is **not** "Paris". If we dismiss the frame, we will lose this information. This is where the *negation function* in frames, δ_{neg} (defined in Section 3) plays its part. Indeed, as discussed before, storing the information that the content of the feature `Location` is "Paris" is done using the *transition function* of the considered frame, adding a transition labelled `Location` linking the node base-labelled "Paris" to the one base-labelled "live" (see Figure 3).

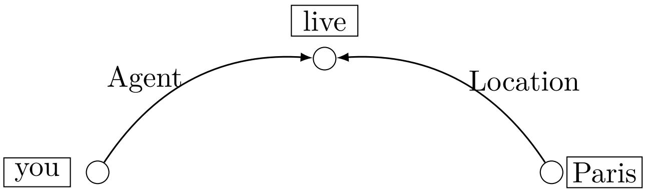


Figure 3: Graphical representation of the frame corresponding to "You live in Paris".

The same idea is now used to store the opposite information, using the *negation function.* We draw a dashed transition labelled `Location` and linking the node base-labelled "Paris" to the one base-labelled "live" (see Figure 4).

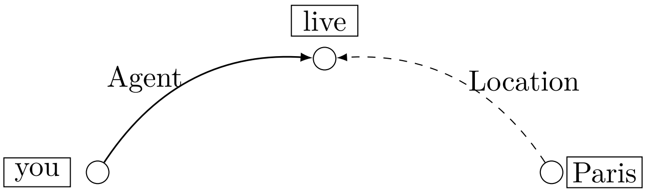


Figure 4: Graphical representation of the frame corresponding to "You don't live in Paris".

Finally, as mentioned above, *yes/no*-questions can also be answered by short answers such as "maybe" or any semantic equivalent. In this case, no information about the content of the frame is given; therefore, we use the *dismiss* operator.
Reaching agreement or disagreement at the end of a negotiation phase is explicit when we consider closed questions. When *wh*-questions (also called open, by opposition to *yes/no*-questions) are involved, the focus of the negotiation phase shifts. Instead of trying to reach a validation of a previously constructed frame, the aim of *wh*-negotiation phases consists in the construction of the frames being discussed.

6 *Wh*-questions

Wh-questions are defined as questions, in English, that give rise to answers whose semantic types match those of the *wh*-phrase contained in the interrogative [Ginzburg and Sag, 2000]. A *wh*-phrase is introduced by a *wh*-word; see Table 1 for a complete list [Aarts et al., 2014].

WH-WORD	QUERY
What	Entity, object
When	Time, moment
Where	Position, place
Who	Person
Whom	Person
Which	Choice, alternative
Whose	Person, entity
Why	Reason
How	Way, manner, characteristic

Table 1: *Wh*-words and corresponding queries.

Focusing on *wh*-questions introduced by is interesting for us as it is gives us the possibility to study the scope of questions introduced by *wh*-words words. Then, determining this scope gives us the possibility to encode the questions and answers in terms of features, therefore frames.

6.1 Feature-based discussions

When we introduced frames, we did not discuss the way the attributes (features) set A should be constructed. The intuition given throughout this work is illustrated by the following example:

Example 11 (*wh*-question)**.**

$\mathbf{A_1}$ *Where do you live?*

$\mathbf{B_2}$ *In Paris.*

Which can then be encoded as:

$$[\![A_1]\!] = \begin{bmatrix} \mathit{LIVE} \\ \texttt{Ag:}\ B \\ \texttt{Loc:}\ ? \end{bmatrix}, \quad [\![B_2]\!] = \begin{bmatrix} \mathit{LIVE} \\ \texttt{Ag:}\ B \\ \texttt{Loc:}\ \mathit{Paris} \end{bmatrix}.$$

Yet, it seems that virtually any set of features could be defined. It is difficult to come up with a set of features which would both be exhaustive, allowing to represent any possible sentence, and computationally realistic – without overlapping and/or redundant scopes of features (one sentence constituent should ideally correspond to one and only one feature). There is a great diversity in features sets one can define. An example of commonly used thematic roles (corresponding to what we designate by features) is presented in [Jurafsky and Martin, 2015], see Table 2.

THEMATIC ROLE	DEFINITION
Agent	The volitional causer of an event. ex: *The waiter* spilled the soup.
Experiencer	The experiencer of an event. ex: *John* has a headache.
Force	The non-volitional causer of an event. ex: *The wind* wrecks the house.
Theme	The participant most directly affected by an event. ex: Brutus stabbed *Caesar.*
Result	The end product of an event. ex: The city built *a house.*
Content	The proposition or content of a propositional event. ex: Mary said "*I met John in the supermarket*".
Instrument	An instrument used in an event. ex: Brutus stabbed Caesar with *a knife.*
Beneficiary	The beneficiary of an event. ex: Mary bought a gift for *John.*
Source	The origin of the object of a transfer event. ex: John flew in from *Paris.*
Goal	The destination of an object of a transfer event. ex: Mary drove to *Warsaw.*

Table 2: Commonly used thematic roles.

A first glance on this list of roles throws us back to the main issue presented before – the non-exhaustivity of such a list. Thinking about Example 6 (Section 4), we see that none of the roles from Table 2 correspond to the features `Direction` and `Location`. `Location` can eventually be viewed as a combination of `Source` and `Goal`, but the same can (with some effort) be said about `Direction`. In our treatment of Example 6, "left" and "here" bore two distinct types of information content, whereas if we directly adopt Jurafsky and Martin's set of features, this distinction will be lost. Yet, boldly adding `Direction` and `Location` to the previous set would still not solve the issue, as `Source`, `Goal`, `Direction` and `Location` would then be redundant. In the following, we chose to build our own set starting from the previous one by pairing, when possible, one *wh*-word to one feature. We will consider the three following groups of *wh*-words, assembled according similarity considerations.

Group 1 Who, Whom, Whose

Group 2 Where, Why, When

Group 3 What, Which, How

6.2 *Wh*-words-based discussions

Group 1

Who According to Table 2, two thematic roles correspond to "Who" – `Agent` and `Experiencer`. For the sake of simplicity and following our pairing rule, we unite these roles under the feature `Agent`.

Whom According to Table 2, two thematic roles correspond to "Whom" – `Theme` and `Beneficiary`. For the sake of simplicity and following our pairing rule, we unite these roles under the feature `Theme`.

Whose As no thematic role can clearly be identified as corresponding to "Whose", we create the role `Owner`.

Group 2

Where As discussed above, the `position, place` awaited as an answer to a question beginning with "Where" can be viewed as objects corresponding to the thematic roles `Source` or `Goal`. As our main aim is to help computability and reduce redundancy, we combine those two thematic roles in one, that we will call `Location`.

Why The word "causer" that can be found both in the definitions of `Agent` and `Force` can be an indication of the fact that those are the thematic roles that should be paired with "Why". As those roles are very unspecific, we chose here to create a new thematic role, identified by the fact that the corresponding sentence constituent should be introduced by "because". We call this role `Reason`.

When It is interesting to notice that Jurafsky and Martin's list does not contain a single role able to represent temporal data. The simplest solution is to create a feature called `Temporality`. The difficulty (that we will not consider here but which should be kept in mind for issues such as frames union) lies in the multiplicity of temporal representations that language can come up with: first, durations can be relative (ex: since 2015) or absolute (ex: for 5 years). Then, one has to distinguish punctual durations (ex: on October the 5th, 2015) from time intervals (ex: in November 2015). On top of that (and this is where unification of features can fail), it is important to notice that a set of punctual durations is not equivalent to a time interval: "every thursday" cannot be considered as a time interval. Yet, it is possible to find punctual durations inside all time intervals (ex: every Thursday of November 2015). Therefore, though we ignore the difficulties here by pairing "When" with `Temporality`, they do not disappear.

Group 3

What The problem of "What" lies in the non-specificity of its usage. In Table 2, an answer to a question starting by "What" could be a `Force`, a `Result`, a `Content` or an `Instrument`. Such a multiplicity of roles is unbearable regarding our pairing task. Yet, these features are so different that uniting them under one designation would badly hurt the expressivity of our model. Therefore, it is necessary, when considering questions introduced by "What", to look at the whole *wh*-phrase contained in the interrogative. Then, the feature corresponding to the answer will be the one semantically typing the focus phrase of the interrogative (ex: "What *time* is it?" corresponds to the feature `Temporality`).

Which This case should be treated in the same way as the previous one. Therefore, "Which + focus phrase" corresponds to the feature of the focus phrase. Ex: "Which city do you prefer, Paris or London?" awaits for an answer of semantic type `Location`.

How The case of "How" is a delicate one, as, similarly to the two previously studied ones, the answers awaited for a question starting by "How" vary greatly depending on the phrase that follows the interrogative word (one can think about the difference between "How much" and "How good"). Yet, unlike for the answers to questions starting by "What" and "Which", it is possible to come up with a unique designation for all the answers to questions starting by "How": we call this feature `Characteristic`. Still, as for `Temporality`, this unique notation only disguises difficulties without dismissing them.

To sum up the discussion above, we present the features we chose with definitions and examples in Table 3.

FEATURE	DEFINITION
Agent	The volitional causer of an event. ex: *The waiter* spilled the soup.
Theme	The participant most directly affected by an event. ex: Brutus stabbed *Caesar*.
Result	The end product of an event. ex: The city built *a house*.
Content	The proposition or content of a propositional event. ex: Mary said "*I met John in the supermarket*".
Instrument	An instrument used in an event. ex: Brutus stabbed Caesar with *a knife*.
Location	The place where an event happens. ex: I live in *Paris*.
Reason	Explanation given for an event. ex: I am happy because *she's getting married*.
Temporality	Time when an event happens. ex: I go to the pool *every thursday*.

Table 3: New set of features.

Finally, keeping in mind the previous discussion on one-to-one *wh*-word to feature pairing, we obtain Table 4.

WORD	FEATURES
What + focus phrase	`Feature`(focus phrase)
When	`Temporality` (`Tmp.`)
Where	`Location` (`Loc.`)
Who	`Agent` (`Ag.`)
Whom	`Theme` (`Th.`)
Which + focus phrase	`Feature`(focus phrase)
Whose	`Owner` (`Ow.`)
Why	`Reason` (`Re.`)
How	`Characteristic` (`Ch.`)

Table 4: *wh*-words and features pairing.

Though we changed the original list of thematic roles presented in Table 2, these changes are not significant enough for us not to be able to use the computability results given in [Jurafsky and Martin, 2015]. Although our model is yet far from being handy in a computational sense, it is still satisfying to check that our theoretical considerations do not drive us too far away from reality.

7 Modeling– SLAM

Our work pursues two purposes – first, provide a way to model dialogue in a correct conversation setting, but also a way to handle yet acknowledge on-going conversational disorders. Throughout our work on the first part of this task, we tested our model on examples from the SLAM corpus, restricting these tests to non-failing conversations first. Yet, even without considering conversational disorders, the real-life nature of these dialogues was enough to make us do some model adjustments.

As the examples and theories presented above seemingly treat only English language, the following will explain in more details our reasons and the way our theories adapt to French. We hereby show few examples coming from the SLAM corpus – to preserve patient's anonymity, the following examples have been anonymized and do not display any sensitive content.

7.1 French interrogatives

One can distinguish two types of French interrogatives: total ones, corresponding to English polar questions, and partial ones, corresponding to English *wh*-ones [Riegel et al., 1994]. Unlike in English, in French, partial questions can be driven by multiple morphological variations of interrogative pronouns and adverbs, which are not linguistically identified as easily as *wh*-words. Table 5 presents correspondences that can be drawn. It was constructed according to the following process:

1. Retrieving of a list of English *wh*-words from [Aarts et al., 2014].
2. Retrieving of a list of French interrogative pronouns and adverbs from [Riegel et al., 1994].
3. Translation of the English set of words to French, using `Reverso`.
4. Translation of the English set of words to French, using `Linguee`.
5. Translation of the French set of words to English, using `Reverso`.
6. Translation of the French set of words to English, using `Linguee`.
7. Compiling the previously obtained information.
8. Verification, using `Systran`.

Now, the theories developed in the previous sections can be applied, giving the French interrogative words to Features correspondence (see Table 5).

WH-WORD	FRENCH EQUIVALENTS	FEATURES
What	que, qu’ quoi, de quoi quel, quelle, quels, quelles	`Feature`(focus phrase)
When	quand	`Temporality` (`Tmp.`)
Where	où	`Location` (`Loc.`)
Who	qui quel, quelle, quels, quelles lequel, laquelle, lesquels, lesquelles	`Agent` (`Ag.`)
Whom	qui lequel, laquelle, lesquels, lesquelles	`Theme` (`Th.`)
Which	qui lequel, laquelle, lesquels, lesquelles auquel, à laquelle, auxquels, auxquelles	`Feature`(focus phrase)
Whose	à qui	`Owner` (`Ow.`)
Why	pourquoi	`Reason` (`Re.`)
How How (much)	en quoi comment combien	`Characteristic` (`Ch.`)

Table 5: English to French correspondences.

It is interesting to notice that though French separates “How much” (« combien ») from the other questions beginning by “How”, the feature-interrogative word pairing stays relevant, as our definition of the features set scopes is broad enough.

7.2 Real-life set-up

Now that everything is set for our theories to be tried on excerpts from the corpus, we will discuss few examples and show the strong sides and the limits of our model.

Example 12 (SLAM)**.**

A_1 *Qu’est ce que vous avez vendu ?*	**A_1** *What did you sell?*
B_2 *Des pulls.*	**B_2** *Sweaters.*

This first small example fits flawlessly in the theory elaborated in Section 6 – the psychologist asks a *wh*-question containing the *wh*-word “What”, followed by the focus phrase “sell”. The feature corresponding to “sell” is `Theme`, so the psychologist’s question interrogates the feature `Theme`. Then, **B** answers, assigning the value “sweaters” to the feature `Theme`. The frame corresponding to the end-product of the computation of this dialogue is:

$$[\![B_2]\!] = \begin{bmatrix} \quad SELL \\ \texttt{Ag:}\ B \\ \texttt{Th:}\ sweaters \end{bmatrix}.$$

The next example is slightly more sophisticated, as it combines phenomena from Sections 5 and 6.

Example 13 (SLAM)**.**

$\mathbf{A_1}$ *Vous habitez où ?*	$\mathbf{A_1}$ *Where do you live?*
$\mathbf{B_2}$ *À T.*	$\mathbf{B_2}$ *In T.*
$\mathbf{A_3}$ *C'est dans la ville de L.*	$\mathbf{A_3}$ *Is it in the city of L?*
$\mathbf{B_4}$ *Oui.*	$\mathbf{B_4}$ *Yes.*

First, the psychologist asks a *wh*-question containing the *wh*-word "Where". That question interrogates the feature `Location`. Then, **B** answers, assigning the value "T" to the feature `Location`. The corresponding frame is then:

$$[\![B_2]\!] = \begin{bmatrix} \mathit{LIVE} \\ \texttt{Ag: } B \\ \texttt{Loc: } T \end{bmatrix}.$$

The conversation continues as the psychologist asks a new question. It is a polar question interrogating the previous frame, specifically the feature `Location`. As the question is about the feature `Location` of the feature `Location` of $[\![B_2]\!]$, we need to construct embedded frames (frames that have more than one typed node), which gives us the following representation for $\mathbf{B_4}$:

$$[\![B_4]\!] = \begin{bmatrix} & \mathit{LIVE} \\ \texttt{Ag: } B & \\ \texttt{Loc: } & \begin{bmatrix} \mathit{IS\ IN} \\ \texttt{Ag: } T \\ \texttt{Loc: } L \end{bmatrix} \end{bmatrix}.$$

Yet, real-life dialogical settings have specificities that are still not fully captured by our model. In the following example, the dialogical rules for turn alternations are not fully observed:

Example 14 (SLAM)**.**

$\mathbf{A_1}$ *Et après vous avez eu...*	$\mathbf{A_1}$ *And then you had...*
$\mathbf{B_2}$ *Ben quand je suis souriante, ça va.*	$\mathbf{B_2}$ *Well, when I'm smiling, it's ok.*

Cognitively, this break in the dialogical rules corresponds to the feeling that **B** is sidestepping the question that the psychologist is about to ask.
An example of the complementary phenomenon is the following:

Example 15 (SLAM)**.**

$\mathbf{A_1}$ *Comment ça se passe, votre quotidien ?*	$\mathbf{A_1}$ *How is your day-to-day life going?*
$\mathbf{B_2}$ *Le quotidien se passe bien.*	$\mathbf{B_2}$ *Day-to-day life is fine.*
$\mathbf{B_3}$ *Le matin je me lève à six heures, je déjeune, je fais ma toilette.*	$\mathbf{B_3}$ *In the morning I wake up at six o'clock, I eat breakfast, I wash myself.*

Here, **B** gives an over-extensive answer to the psychologist's question. This answer is composed of two parts: a short, closed answer, satisfying the question, and a second, elaborated answer. Two interpretations are possible here – first, one can consider that $\mathbf{B_2}$ and $\mathbf{B_3}$ are each independent answers to $\mathbf{A_1}$ and should be treated as such. In this case, the main problem will be in the choice of the computation that should be done to obtain the final representation of this conversation – should we compute the union of the two final frames? should we only keep one of them, and if yes, which one?
The second idea would be to say that (following the terminology given in [Asher and Lascarides, 2003]) $\mathbf{B_3}$ is an elaboration of $\mathbf{B_2}$. Yet, in this case, our frame model should be enriched with rhetorical relations. Therefore, the most satisfying solution that we can suggest for now is the use of embedded frames.

7.3 Observations

Thinking back to the dialogical context considerations developed in Section 5, it is important to notice that the operations *dismiss* and *store* can be triggered in configurations outside polar questions. In the following example, the storage operation should be triggered by the utterance "OK." (« D'accord. »).

Example 16 (SLAM)**.**

$\mathbf{A_1}$	*Vous êtes arrivés pour quoi au V. ?*	$\mathbf{A_1}$	*For what reason did you come to V.?*
$\mathbf{B_2}$	*Pour une TS.*	$\mathbf{B_2}$	*For a suicide attempt.*
$\mathbf{A_3}$	*D'accord.*	$\mathbf{A_3}$	*OK.*

Indeed, after **B**'s answer, the psychologist does not have anything to add on the topic. « D'accord » can here be interpreted as a linguistical expression of the end of a negotiation phase of the dialogue. As shown above, « D'accord » is directly translatable in English without loss of meaning – according to `Reverso`, accurate translations are either "All right" or "OK" and `Linguee` gives "OK" as main translation and "All right" as a less used variant.
That is why it is interesting to look at the following example:

Example 17 (SLAM)**.**

$\mathbf{A_1}$	*Avoir mal physiquement on le sait depuis tout petit ce que ça fait*	$\mathbf{A_1}$	*From a young age, you know what it feels like to be hurt physically*
$\mathbf{B_2}$	*Quand on tombe on se fait un bleu ?*	$\mathbf{B_2}$	*When you fall, you get a bruise?*
$\mathbf{A_3}$	*Voilà.*	$\mathbf{A_3}$	*Yes, that's it.*

The semantic phenomenon in the original excerpt is the same here as in the previous example. However, any direct translation attempt fails – `Reverso` produces the set "here is; well; so; that's it; now", while `Linguee` adds "that is" and "there". The English translation of « Voilà » presented in Example 18 (cross-validated by an American English native-speaker) attempts to translate the utterance largely taking into account its context. Still, it under-specifies the meaning and transforms the observed phenomenon, shifting it from a linguistical expression of the end of a negotiation phase of the dialogue to an enhanced short answer to an implied polar question. Yet, modeling both leads to a triggering of the storage operation – which illustrates the positive side of not diving too deep in linguistic details while defining a model.

Conclusion

Dialogue modeling is a computational semantics task that inherits methods and insights from semantics of discourse. Aiming to design our model in a way that would fit data as specific as the SLAM corpus has challenged us into defining a new framework in the field of semantics of dialogue by combining TTDL, a discourse formalism, with a formal view of Frame Semantics which provide us with a controlled way of storing information. I targeted precise theoretical points lying in definitions of frames' formalism in order to strengthen the model in regards of compositionnality. I was then able to check that this formalization is compatible with the intuitions given in [Tiv, 2016] and conversely that those intuitions used properties that could be proved (such as Proposition 2).

This view of frames allows a proper use of embedded frames, helping to extend the compositional properties of the model, which happen to be an issue as soon as the dialogues get bigger than toy ones. As the SLAM corpus' most encountered dialogical moves are questions and answers, I specifically focused on a methodical study of these phenomena. Finally, I confronted our model to real-life data, unveiling its strong compositional and question-answering phenomena handling sides and raising new interrogations and challenges to be tackled in future developments.

In particular, the ultimate aim of this work is to forge a model of dialogical interaction strong enough to both handle correct conversation and acknowledge conversational failures. In a correct dialogical setting, applying this model should result in a non-contradictory logical proposition, whereas a conversational failure should be handled differently. Consider the example bellow:

Example 18 (SLAM).

A$_1$ *Oh ouais et pis compliqué et c'est vraiment très très compliqué la politique c'est quelque chose quand on s'en occupe faut être gagnant parce qu'autrement quand on est perdant c'est fini quoi*

B$_2$ *Oui*

A$_3$ *J. C. D. est mort, L. est mort, P. est mort euh (...)*

B$_4$ *Ils sont morts parce qu'ils ont perdu à votre avis*

A$_5$ *Non ils gagnaient mais si ils sont morts, c'est la maladie quoi c'est c'est*

A$_1$ *Oh and it's complicated and politics is really very complicated, it's something when you do that, you need to win because otherwise when you lose it's the end you know*

B$_2$ *Yes*

A$_3$ *J. C. D. is dead, L. is dead, P. is dead hum (...)*

B$_4$ *They died because they lost, according to you*

A$_5$ *No, they were winning but if they died, it was out of sickness you know it's it's*

Though no dialogical rule has been broken, this example does not feel right. Indeed, **A** operates an unconscious semantic shift in the usage of the word « mort » ("dead") allowing both senses to be accessible in the rest of the dialogue in a blurry way. **A** does not stick to one sense inside one negotiation turn of the dialogue, and **B** has to ask clarifying questions.

This type of semantic shifting is characteristic of schizophrenics – see [Rebuschi et al., 2014] for further insight on the subject. It is therefore at the core of what future extensions of our model should be able to recognize, acknowledge and handle. Our future work will be to strengthen and enrich the model presented in this master thesis, stretching it in order to be able to add epistemic logic considerations in the dialogical context. Dynamic epistemic modal logic provides a way storing information and reasoning on knowledge (see [Van Ditmarsch et al., 2007]), and using it in our model would give us the possibility to add new storage operations and limit cognitive information loss.

Acknowledgements

I would like to thank my internship advisor Maxime Amblard for directing my research and patiently guiding me through a field of computational linguistics that was thoroughly new to me. I would also like to thank the whole SÉMAGRAMME team – Philippe De Groote, Bruno Guillaume, Nicolas Lefebvre, Guy Perrier and Sylvain Pogodalla for the various help and advice they offered me. Finally, I particularly thank my trying student-room-mates: Clément Beysson and Mathieu Huot, without whom this work would not have been written, and Sandro Maskharashvili for his wisdom and insight.

References

[Aarts et al., 2014] Aarts, B., Chalker, S., and Weiner, E. (2014). *The Oxford dictionary of English grammar*. Oxford University Press.

[Amblard et al., 2015] Amblard, M., Fort, K., Demily, C., Franck, N., and Musiol, M. (2015). Analyse lexicale outillée de la parole transcrite de patients schizophrènes. *Traitement Automatique des Langues*, 55(3):91 – 115.

[Asher and Lascarides, 2003] Asher, N. and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

[Church, 1940] Church, A. (1940). A formulation of the simple theory of types. *The journal of symbolic logic*, 5(02):56–68.

[de Groote, 2006] de Groote, P. (2006). Towards a montagovian account of dynamics. *Proceedings of semantics and linguistic theory XVI*.

[Fillmore, 1982] Fillmore, C. (1982). Frame semantics. *Linguistics in the morning calm*, pages 111–137.

[Fouqueré and Quatrini, 2012] Fouqueré, C. and Quatrini, M. (2012). Un cadre formel issu de la théorie de la démonstration pour la théorie de l'argumentation. *Mathématiques et sciences humaines. Mathematics and social sciences*, (198):49–83.

[Ginzburg, 2016] Ginzburg, J. (2016). Semantics of dialogue. *The Cambridge Handbook of Formal Semantics*.

[Ginzburg and Sag, 2000] Ginzburg, J. and Sag, I. (2000). *Interrogative investigations*. Stanford: CSLI publications.

[Grice, 1975] Grice, H. P. (1975). Logic and conversation. *Syntax and Semantics*, page 3:41–58.

[Hamblin, 1970] Hamblin, C. L. (1970). *Fallacies*. Methuen, London.

[Heim, 1982] Heim, I. (1982). The semantics of definite and indefinite noun phrases.

[Jurafsky and Martin, 2015] Jurafsky, D. and Martin, J. H. (2015). *Speech and Language Processing*.

[Kallmeyer and Osswald, 2014] Kallmeyer, L. and Osswald, R. (2014). Syntax-driven semantic frame composition in lexicalized tree adjoining grammars. *Journal of Language Modelling*, 1(2):267–330.

[Kamp, 1981] Kamp, H. (1981). A theory of truth and semantic representation. *Formal semantics – the essential readings*, page 189–222.

[Kamp and Reyle, 2013] Kamp, H. and Reyle, U. (2013). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.

[Lecomte et al., 2007] Lecomte, A., Quatrini, M., and Fleury-Donnadieu, M.-R. (2007). Une introduction à la ludique et à ses applications à la pragmatique.

[Montague, 1973] Montague, R. (1973). The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer.

[Plüss, 2014] Plüss, B. (2014). *A Computational Model of Non-Cooperation in Natural Language Dialogue*. PhD thesis, The Open University.

[Rebuschi et al., 2014] Rebuschi, M., Amblard, M., and Musiol, M. (2014). Using sdrt to analyze pathological conversations: Logicality, rationality, and pragmatic deviances. In *Interdisciplinary Works in Logic, Epistemology, Psychology and Linguistics*, pages 343–368. Springer.

[Riegel et al., 1994] Riegel, M., Pellat, J.-C., and Rioul, R. (1994). Grammaire méthodique du français. *Linguistique nouvelle.*

[Schlöder et al., 2016] Schlöder, J. J., Venant, A., and Asher, N. (2016). Aligning intentions: Acceptance and rejection in dialogue. *SEMDIAL 2016 JerSem*, page 118.

[Stalnaker, 2002] Stalnaker, R. (2002). Common ground. *Linguistics and philosophy*, 25(5-6):701–721.

[Tiv, 2016] Tiv, S. (2016). Modeling dialogue in dynamic framework. Master's thesis, Université de Lorraine.

[Van Ditmarsch et al., 2007] Van Ditmarsch, H., van Der Hoek, W., and Kooi, B. (2007). *Dynamic epistemic logic*, volume 337. Springer Science & Business Media.

[Vosoughi and Roy, 2016] Vosoughi, S. and Roy, D. (2016). Tweet acts: A speech act classifier for twitter. *arXiv preprint arXiv:1605.05156.*

[Walton, 2003] Walton, D. (2003). The interrogation as a type of dialogue. *Journal of Pragmatics*, 35(12):1771–1802.

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations.* Basil Blackwell, Oxford.